

In the claims:

Please substitute the following full listing of claim for the claims as originally filed or most recently amended.

1. (Currently Amended) An instruction buffer comprising:

a sequence of instructions arranged in an order determined beforehand; and

a first buffer including entries arranged in a preselected entry number order for storing said sequence of instructions; and

a second buffer including other entries for storing instructions, wherein an instruction stored in any one of said other entries earlier than other instructions is issued earlier than said other instructions,

wherein any one instruction of said sequence of instructions stored in any one of the entries designated by a low relatively lower entry number is prior, in order, to another instruction stored in another entry designated by a high relatively higher entry number.

2. (Original) The instruction buffer as claimed in claim 1, wherein the entries each show whether or not the instruction stored therein is ready to be issued.

3. (Currently Amended) The instruction buffer as claimed in claim 2, wherein the instruction ~~is~~ first issued from[[],] among the entries whose instructions are ready to be issued[[],] is the entry having a lowest entry number among said entries whose instructions are ready to be issued.

4. (Original) The instruction buffer as claimed in claim 3, wherein the entries storing the instructions are lower in entry number than the entries storing no instructions.

5. (Cancelled)

6. (Currently Amended) A method of controlling a buffer queue, comprising the steps of:

generating a first group of instructions in ~~an a~~ priority order determined beforehand;

generating a second group of instructions belonging to said first group of instructions and capable of being executed; and

executing one instruction of said second group of instructions highest in priority order.

7. (Currently Amended) The method as claimed in claim 6, further comprising the steps of:

generating a third group of instructions included in said first group of instructions; and

generating a fourth group of instructions included in said first group of instructions and not dependent on said third group of instructions;

wherein when one of said fourth group of instructions highest in priority order does not belong to said second group of instructions, ~~none no~~ instruction of said fourth group of instructions is executed.

8. (Currently Amended) The method as claimed in claim 7, wherein one of ~~preselected~~ two instructions belonging to said third group or fourth group of instructions is not executable until the other instruction of said two instructions is executed.

9. (Original) The method as claimed in claim 8,
wherein the instructions belonging to said third group
are executed at the same time as the instructions
belonging to said fourth group.

10. (Original) The method as claimed in claim 9,
wherein the instructions belonging to said third group
and the instructions belonging to said fourth group and
operation instructions and memory access instructions,
respectively.